iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63398

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Boost Software Quality with Devops Practices

Shubham Patil[1], Satish Patil[2], Prof. Pavan Mitragotri[3]
[1, 2]*Post Graduate Student,* [3]*Professor, Department of Master of Computer Applications, KLS Gogte Institute of Technology, Belagavi, India*

Abstract: ***DevOps, an extension of agile methods, integrates numerous principles to improve communication between development and operations teams. The fundamental goal of this research is to look at the effects of DevOps adoption on software quality, as well as to explore effective ways for improving quality effectively. An thorough literature investigation was conducted to investigate existing DevOps methods in the sector, resulting in the construction of a conceptual research model with five derived hypotheses. The study objectives were met by evaluating hypotheses with Pearson correlation and developing a linear model based on linear regression analysis. To collect quantifiable data, an online questionnaire was used, followed by interviews with DevOps and Quality Assurance professionals to determine how DevOps may improve software quality. Feedback from interviews, coupled with The hypotheses were tested using regression analysis, and suggestions were based on them. The quantitative study's findings show that adopting DevOps leads to considerable improvements in software quality, particularly when using the CAMS (Culture, Automation, Measurement, Sharing) architecture. Among the considerations, automation emerged as the most important component for improving software quality. The findings of multiple regression analysis highlight the relevance of culture, automation, measurement, and sharing in the goal of higher software quality. In conclusion, this study strongly argues for the use of DevOps to produce high-quality software. Organizations may significantly improve the overall quality of their software products by implementing the CAMS Framework and adhering to concepts such as automation.***
Keywords: ***DevOps, CAMS Framework, Quality, ISO 9126, Automation.***

I. INTRODUCTION

Software has developed into a vital component of enterprises during the last 20 years. Innovative languages, software architectures, development tools, and technologies including cloud computing, crowdsourcing, application programming interfaces (APIs), rest services, big data, and the internet of things (IoT) have all been presented by software engineering researchers on a constant basis. Software development process models are used in the IT industry as a standard framework to organize, schedule, and manage the effective and productive development of information systems. For this reason, a number of software development life cycle models have been established and created. Every model adheres to a distinct set of guidelines designed to guarantee software development success. The Big-Bang, V-, Spiral, Iterative, and Waterfall models are a few examples of conventional models. The Agile technique was first presented in 2001. It resulted in the adoption of ideas like Kanban and Scrum. In recent times, DevOps has surfaced as an enhanced iteration of Agile, emphasizing operational elements.

In order to produce software and services quickly, dependable, and of the highest caliber, DevOps encourages developers and operations teams to work together and communicate seamlessly. Combining the terms "Dev" from developers and "Ops" from operations results in the name "DevOps." Throughout the whole lifespan of the service, from development to deployment and support, it gives teams shared accountability and obligations. The continuous integration and release objectives of Agile are extended to a DevOps environment by emphasizing cross-functionality, shared responsibility, and trust. In order to handle frequent releases, automation of configuration, release, and modification procedures is recommended.

The purpose of this study is to ascertain whether the Software organizations that use DevOps see an improvement in software quality. The report highlights important elements that should be taken into account to improve software quality and shows businesses how to use DevOps successfully.

II. BACKGROUND AND RELATED WORK

A. SDLC.

The Software Development Life Cycle (SDLC) is a conceptual model that describes the process of planning, creating, testing, and deploying software. The industry employs a variety of approaches to properly manage the SDLC. Agile is a well-known approach that emerged in reaction to the limits of inflexible plan-driven process models such as Waterfall and Rational Unified Process (RUP), among others. Agile was also created to solve the issues raised by earlier resistance approaches.

Agile is a novel method to project management that is largely used in software development projects. It stresses iterative and incremental development, which encourages flexibility and adaptation in the development process. Unlike traditional plan-driven approaches, Agile promotes cooperation and continual customer input, resulting in superior outcomes. and client happiness.

Agile allows software development teams to adapt fast to changing needs, provide useful features gradually, and retain a focus on high-quality software. Agile has become a popular technique in the software industry because to its dynamic and customer-centric approach, which helps to produce successful software solutions. The software development sector has seen a tremendous increase characterized by help address inefficiencies in their current procedures. Notable IT companies like as IBM, Facebook, and Microsoft have used Continuous Deployment to simplify their development operations. Continuous Deployment, as defined by Claps, Svensson, and Aurum, is a difficult task because to the large influence on system stability. Nonetheless, it creates new commercial prospects while presenting innovative obstacles to software firms. Continuous Deployment is not without challenges, including careful consideration of system stability and rigorous testing techniques. However, the benefits of speedier and more streamlined development cycles, as well as increased customer satisfaction, have encouraged these organizations to investigate and use this strategy.

unpredictability and quick change. These modifications are generally driven by changing customer requirements and favorable responses to change requests. To satisfy these dynamic needs and assure improved customer satisfaction, several businesses have adopted agile development approaches, which enable rapid releases and responsiveness. While the majority of firms have adopted agile techniques, others have gone farther and implemented specialized agile practices.

B. DevOps

"DevOps is a set of principles designed to improve collaboration between development and operations teams." It aims to address common objectives, incentives, procedures, and tools in order to bridge any gaps that may naturally emerge between diverse groups. While total alignment of shared objectives and incentives is not always possible owing to inherent conflicts, they should be harmonized with one another." The fundamental goal of DevOps is to identify and eliminate differences between the Development and Operations teams. Professionals working in software development frequently approach their responsibilities with an emphasis on providing product updates. The major DevOps goals, as identified in, are as follows:

- Provide demonstrable business value through high-quality service delivery.
- Prioritize simplicity and agility throughout. issues like as technology, procedures, and human factors.
- Remove barriers between development and operations by building trust, shared ownership, and an innovative and collaborative culture.
- Manage dynamic compliance while keeping up with changing rules and regulations governing access and data sharing.

The Phoenix Project has outlined the divisions that may be crossed when using DevOps, which include Security Compliance and IT, Development and Quality Assurance, Marketing and IT, and Business Goals and IT Delivery. Behr and Kim of The Phoenix Project underline that DevOps does not guarantee flawless results with regular delivery, robust platforms, and full business-IT alignment. Instead, the emphasis should be on building a collaborative atmosphere in which everyone can give their all to solve problems and support the business. According to the State of DevOps According to the 2016 report, applying DevOps enhanced organizational performance, sales, and profitability. The use of DevOps methods has grown, with the number of DevOps teams increasing from 16% in 2014 to 19% in 2015 and 22% in 2016. High-performing organizations have dramatically improved their deployment frequency, resulting in shorter lead times. They beat low-performing companies by delivering changes 200 times more frequently and with 2,555 times quicker lead times. Notably, huge corporations such as Amazon and Netflix have shown the efficacy of DevOps by releasing updates hundreds of times every day. These convincing data demonstrate why businesses are increasingly adopting DevOps as a critical technique in their software development operations.

C. CAMS Framework.

The CAMS Model was created by DevOps pioneers John Willis and Damon Edwards to symbolize the essential concepts of Culture, Automation, Measurement, and Sharing. This paradigm is widely accepted by DevOps practitioners as the foundational set of rules for effective DevOps deployment.

Culture is the most important component of any DevOps deployment. It highlights the need of collaboration among teams, with everyone sharing duties for the end users of their product. DevOps fosters a collaborative culture, which not only supports the adoption of agile approaches in operations, but also allows for information sharing between developers and operations teams, thereby breaking down conventional barriers.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Automation is critical to enable the adoption of DevOps. The correct balance of people, procedures, and tools is critical toDevelop an automated framework for DevOps processes. Operations and testing teams frequently provide useful insights into application performance, and they should educate developers on the need of sustaining application performance in large-scale systems with high loads. A common performance language may be built by implementing automated performance monitoring tools across all environments, from continuous integration and testing to production deployment. A successful DevOps strategy relies heavily on measurements such as business key performance indicators, system metrics, and application behaviour. These metrics should be open, transparent, accessible, meaningful, and readily viewed by all DevOps team members. By resolving performance issues early in the testing process, performance engineers on testing teams may focus on completing large-scale load tests in production-like scenarios, assuring reliable performance.

SharingThe tools, insights, and lessons learnt are crucial to the success of any organization's DevOps deployment [7]. Collaboration thrives on idea sharing and issue solving, which is key to the DevOps mindset. DevOps emphasizes openness and transparency, creating an atmosphere where sharing knowledge and insights is highly appreciated.

III. RESEARCH MODEL

Based on a thorough literature study and the primary research goal, the researchers developed the conceptual framework depicted in Figure 1. The framework is broken into two sections that cover the key variables for DevOps deployment and its influence on software quality.

The hypotheses developed in this study aim to investigate the correlations between variables in accordance with the primary and secondary objectives. The following hypothesis were derived:

1) Hypothesis 1: Implementing DevOps improves software quality.

H10: No meaningful association exists between DevOps implementation and software quality.

H1A: Implementing DevOps has a major impact on software quality.

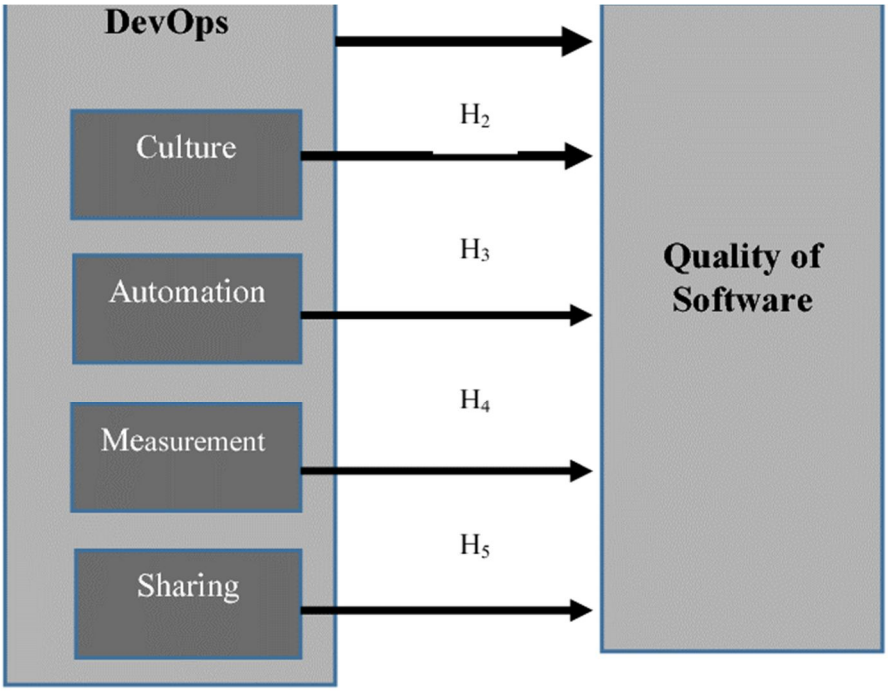


Fig. 1. Research Model and research hypotheses [1]

2) Hypothesis 2: The DevOps culture has a favorable impact on software quality.

H20: There is no substantial association between the DevOps culture and software quality.

H2A: There is a considerable association between DevOps culture and software quality.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

3) Hypothesis 3: Automation in DevOps improves software quality.
H30: Automation in DevOps has no meaningful association with software quality.
H3A: Automation in DevOps has a substantial impact on software quality.

4) Hypothesis 4: Sharing in DevOps improves software quality.
H40: Sharing in DevOps has no meaningful association with software quality.
H4A: There's is a considerable association between sharing in DevOps and software quality.

5) Hypothesis 5: Measurement in DevOps improves software quality.
H50: There is no substantial association between DevOps measurements and software quality.
H5A: Measurement in DevOps has a substantial impact on software quality.

To verify these assumptions and assess the variables "Improve software quality through DevOps" and "Software quality," a carefully constructed online questionnaire was issued to software experts working in DevOps-enabled firms. The questionnaire comprised both five-point Likert scale questions and open-ended questions to elicit comprehensive information. To gather more information, face-to-face interviews were done with DevOps specialists.

The independent variable" Practice DevOps" was assessed using the criteria of Culture, Automation, Measurement, and Sharing. The study's dependent variable was "quality of the software." Indicators for each

TABLE1: OPERATIONALIZATION TABLE

Variables	Indicators	Source
Software Quality	1. Functionality	[9]
	2. Reliability	
	3. Efficiency	
	4. Maintainability	
	5. Usability	
	6. Portability	
Practice DevOps	1.Culture	[7]
	2. Automation	
	3. Measurement	
	4. Sharing	

. variable were identified based on existing literature sources, as depicted in Table 1

After creating the questionnaire, the researchers ran a pilot study with 15 participants. To confirm the internal consistency of the measured variables, the statistical method Cronbach's Alpha was used, with questions having a Cronbach's alpha greater than 0.7.

In Sri Lanka, there are roughly 200 registered software development enterprises divided into three categories: small, medium, and big. Respondents were chosen based on their DevOps experience, particularly if they had previously practiced DevOps.

The researchers' goals for data analysis were to assess data quality and support the hypotheses they had formulated. Graphical statistical methods such as the normality curve and boxplot diagrams were used to evaluate the data's normal distribution. After it was established that all variables showed a normal Pearson correlation was utilized to determine the correlations between the independent and dependent variables.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

IV. RESULTS AND ANALYSIS

A. Quantitative data analysis and hypothesis testing:

An online questionnaire was given to over 300 software experts that work in DevOps firms and have prior DevOps experience. Out of the replies submitted, 120 were chosen for further investigation depending on whether the respondents had prior DevOps expertise and their professional background.

Figure 2 demonstrates the respondents' DevOps experience. The majority, 62%, have two to three years of experience with DevOps. Approximately 24% have less than one year of DevOps experience, while 14% are professionals who have been practicing DevOps for more than three years.

Figure 3 depicts the development process adopted by respondents at their different companies. The majority (66%) comes from companies that practice both. DevOps and Agile methodologies In terms of software development technique, 24% of respondents worked for organizations that used Agile. Interestingly, 10% of respondents have prior DevOps expertise, while working in businesses that use the conventional waterfall process.

Figure 4 shows the distribution of respondents according to their classifications or tracks. The majority, 45%, are on the Quality Assurance track. 21% come from the development track, 23% from operations, and the remaining 11% from project management.

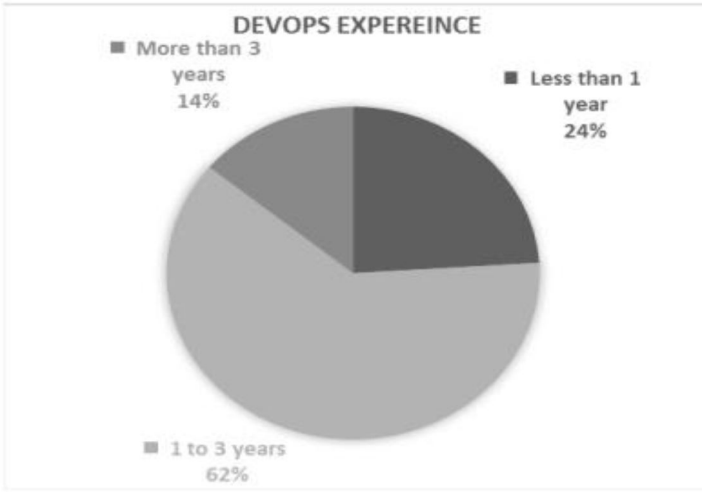


Fig. 2. DevOps Experience

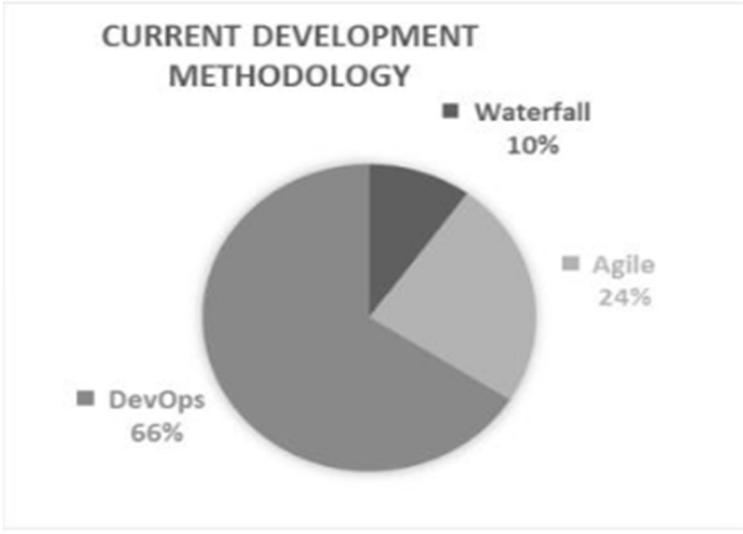


Fig. 3. Current Development Methodology

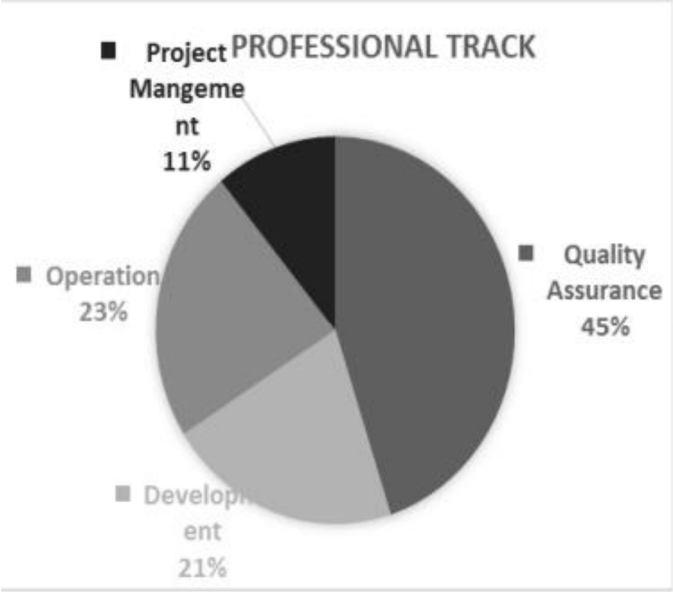


Fig. 4. Professional Track

Table 2 presents a summary of the correlation analysis. At the 99% confidence level, the dependent and independent variables have substantial positive correlations. Based on these findings, all five hypotheses were verified, indicating a substantial positive link between the dependent and independent variables.

TABLE 2: PEARSON CORRELATION

		Quality of Software	Relationship
Practice DevOps	Pearson correlation	0.76**	Strong
	Sig. (2-tailed)	0	
Culture	Pearson correlation	0.741**	Strong
	Sig. (2-tailed)	0	
Automation	Pearson correlation	0.758**	Strong
	Sig. (2-tailed)	0	
Measurement	Pearson correlation	0.704**	Strong
	Sig. (2-tailed)	0	
Sharing	Pearson correlation	0.717**	Strong
	Sig. (2-tailed)	0	
**Correlation is significant at the 0.01 level (2-tailed)			

The quality of the program or product is the most important aspect in the IT company' success. The major goal of this study was to investigate and determine the link between DevOps methods and their influence on software quality. To achieve this purpose, a multiple regression analysis was used to create a model that correctly depicts the link between DevOps and Quality. The equation is expressed in the following notation:

Quality equals β0 + β1C + β2A + β3M + β4S.

Where:

Quality is the dependent variable, representing total program quality.

C indicates Culture, an independent variable that measures the impact of DevOps culture on software quality.

A stand for Automation, an independent variable that measures the influence of DevOps automation strategies on software quality.

M signifies Measurement, an independent variable that represents the impact of DevOps measurement procedures on software quality.

S symbolizes Sharing, an independent variable that indicates how DevOps sharing methods affect software quality.

Using this multiple regression analysis and the resulting equation, the research intends to shed light on the link between DevOps practices and software quality, supporting IT firms in understanding DevOps' critical role in reaching greater levels of software excellence.

$$\text{Software Quality} = 1.409 + 0.176(C) + 0.227(A) + 0.096(M) + 0.172(S)$$

B. Recommending

The major goal of this research is to uncover effective ways for improving software quality in a DevOps setting. To do this, qualitative questionnaires and interviews were performed with DevOps and Quality Assurance professionals, taking into account elements like as culture, automation, measurement, and sharing.

Quality is an important factor in customer happiness, and using a DevOps strategy is key to improving software quality. The research findings highlight the importance of automation as a critical success element in quality improvement. To maintain long-term viability, software teams must carefully examine the Return on Investment (ROI) before moving further with automation.

To successfully use DevOps, businesses must find or establish a team with automation and technical expertise and give suitable training to improve their automation capabilities. When automation resources are rare, hiring workers with the necessary abilities becomes a critical action item.

DevOps best practices include Test Driven Development (TDD), Behavior Driven Development (BDD), and Acceptance Test Driven Development. TDD begins with creating tests before coding, whereas BDD entails collaborating with business stakeholders to explain desired functionality in normal language, and ATDD focuses on identifying scenarios from the end-user's perspective. Before deploying automation, it is critical to create a quality environment. In the DevOps framework, several technologies may be utilized efficiently, including Jenkins for Continuous Integration, Cucumber for BDD, Junit for TDD, GIT for configuration management, Quality Centre, JIRA, ALM for Test lifecycle and defect management, Selenium, QTP, and UFT for automation.

DevOps approaches rely heavily on Continuous Integration (CI). Automated scripts should be performed and implemented into the CI environment to assess the build's stability and status.

Culture has an important role in team cooperation and responsibility sharing. Encourage reciprocal interchange between Development and Operations teams to tear down barriers and promote a culture of learning from one another. Management should foster an environment in which all employees speak the same language and collaborate to solve present problems, avoiding finger-pointing and determining fundamental causes. Performance engineers should focus on large-scale load tests in production-like conditions, working closely with Operations. Sharing environments, frameworks, and scripts between teams can help save time and effort.

To summarize, DevOps approaches may dramatically increase software quality, but they require a focus on aspects such as automation, collaboration, and a culture of shared accountability. Proper training, tool selection, and continuous integration are all necessary for success.

V. CONCLUSIONS

The primary objective of this paper is to conduct an analysis to determine whether the implementation of DevOps leads to an improvement in software quality. To achieve this, data was gathered through online questionnaires and interviews with DevOps experts in the software development industry.

After collecting and analysing the data, the research revealed a positive relationship between the practice of DevOps and the quality of software. It was observed that software quality tends to increase when DevOps is implemented. Furthermore, there was a strong positive correlation between Culture, Automation, Measurement, and Sharing with software quality. This implies that practicing automation and sharing knowledge positively influences software quality.

The findings of this research provide valuable insights for companies that practice DevOps and quality engineering teams, enabling them to make informed decisions to enhance their testing practices. The research results clearly indicate that factors such as culture, automation, measurement, and sharing have a significant impact on the quality of products. Therefore, by correctly considering these factors and practicing DevOps, companies can improve software quality.

VI. ACKNOWLEDGMENT

This research was made possible through the contributions and support of numerous individuals and organizations. First and foremost, I would like to express my sincere gratitude to the DevOps and Quality Assurance professionals who participated in the online questionnaires and interviews. Their insights and expertise were invaluable to the depth and success of this study.

I would like to extend my heartfelt thanks to my academic advisor, [Pavan Mitragotri], for their guidance, encouragement, and constructive feedback throughout the research process. Their knowledge and support were instrumental in shaping the direction and quality of this work.

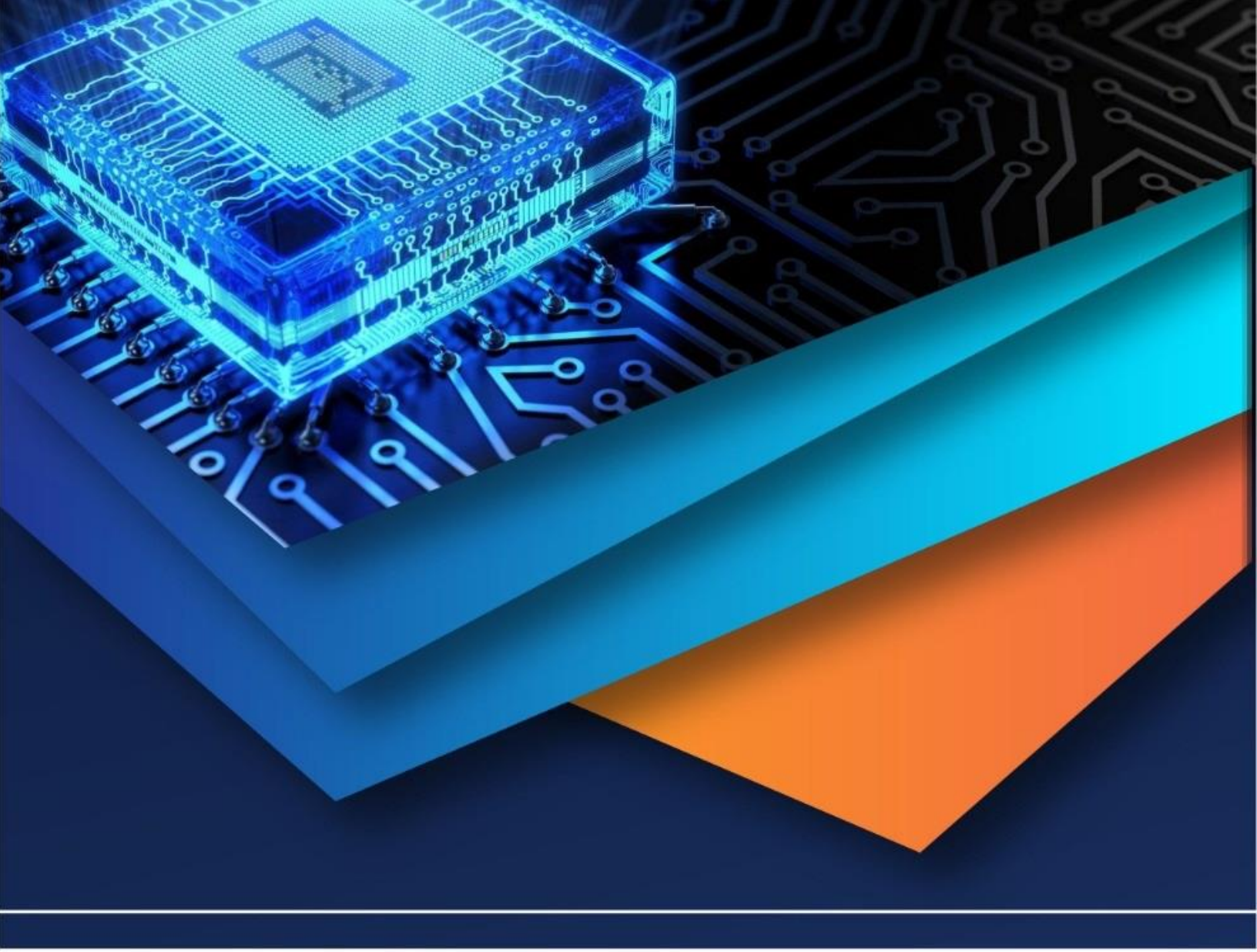
I am also grateful to my colleagues and peers who provided ongoing support and insightful discussions, which significantly enriched the research. Special thanks go to [*Department of Master of Computer Applications, KLS Gogte Institute of Technology*] for providing the necessary resources and a conducive environment for conducting this research.

Additionally, I would like to acknowledge the contributions of the numerous authors and researchers whose work laid the foundation for this study. Their dedication to advancing the field of DevOps and software quality is deeply appreciated.

Thank you all for your invaluable contributions

REFERENCES

[1] Maximilien De Bayser, Renato Cerqueira, Leonardo Guerreiro Azevedo "ResearchOps: The case for DevOps in scientific applications" Article June 2015
[2] Pratibha Jha, Rizwan Khan "A Review Paper on DevOps: Beginning and More To Know", International Journal of Computer Applications (0975 – 8887) Volume 180 – No.48, June 2018
[3] M. Huttermann, "Beginning DevOps for Developers," in DevOps for Developers, 2012
[4] D. L. Farroha and B. S. Farroha, "A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment," In Military Communications Conference, 2014.
[5] Ravi Teja Yarlagadda, "How DevOps Enhances the Software Dévelopment Quality", Article in SSRN Electronic Journal · August 2019
[6] Floris Erich, Chintan Amrit, "A Qualitative Study of DevOps Usage in Practice" ,June 2017
[7] D. Farley and J. Humble, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, 2010.
[8] Georges Bou Ghantous, Asif Qumer Gill, " DevOps: Concepts, Practices, Tools, Benefits and Challenges", Summer 7-19-2017
[9] Dhaya Sindhu Battina , "Devops, A New Approach To Cloud Development & Testing ", 2020 JETIR August 2020, Volume 7, Issue 8
[10] Capemgini and Sogeti, "DevOps with Quality," 2016. [Online]. Available:https://www.sogeti.com/globalassets/global/downloads/testing/pov_devops-with-quality_ok.pdf
[11] Alok Mishra, Ziadoon Otaiwi,"DevOps and software quality: A systematic mapping", November 2020.
[12] Indika Perera, " Improve software quality through practicing DevOps", Conference Paper · September 2017
[13] Lakshmisri Surya, "AI and DevOps in information technology and its future in the United States", 2 February 2021



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY